HHS Public Access

Author manuscript

Pattern Recognit. Author manuscript; available in PMC 2020 February 01.

Published in final edited form as:

Pattern Recognit. 2019 February ; 86: 188–200. doi:10.1016/j.patcog.2018.09.007.

Sparse Autoencoder for Unsupervised Nucleus Detection and Representation in Histopathology Images

Le Hou[1], **Vu Nguyen**[1], **Ariel B. Kanevsky**[1,2], **Dimitris Samaras**[1], **Tahsin M. Kurc**[1,3,4], **Tianhao Zhao**[3,5], **Rajarsi R. Gupta**[3,5], **Yi Gao**[6], **Wenjin Chen**[7,8], **David Foran**[7,8,9], and **Joel H. Saltz**[1,3,5,10]

[1]Dept. of Computer Science, Stony Brook University, Stony Brook, NY, USA

[2]Montreal Institute for Learning Algorithms, University of Montreal, Montreal, Canada

[3]Dept. of Biomedical Informatics, Stony Brook University, Stony Brook, NY, USA

[4]Oak Ridge National Laboratory, Oak Ridge, TN, USA

[5]Dept. of Pathology, Stony Brook University Medical Center, Stony Brook, NY, USA

[6]School of Biomedical Engineering, Health Science Center, Shenzhen University, China

[7]Center for Biomedical Imaging & Informatics, Rutgers, the State University of New Jersey,New Brunswick, NJ, USA

[8]Rutgers Cancer Institute of New Jersey, Rutgers, the State University of New Jersey, NJ, USA

[9]Div. of Medical Informatics, Rutgers-Robert Wood Johnson Medical School, Piscataway Township, NJ, USA

[10]Cancer Center, Stony Brook University Hospital, Stony Brook, NY, USA

Abstract

We propose a sparse Convolutional Autoencoder (CAE) for simultaneous nucleus detection and feature extraction in histopathology tissue images. Our CAE detects and encodes nuclei in image patches in tissue images into sparse feature maps that encode both the location and appearance of nuclei. A primary contribution of our work is the development of an unsupervised detection network by using the characteristics of histopathology image patches. The pretrained nucleus detection and feature extraction modules in our CAE can be fine-tuned for supervised learning in an end-to-end fashion. We evaluate our method on four datasets and achieve state-of-the-art results. In addition, we are able to achieve comparable performance with only 5% of the fully-supervised annotation cost.

Keywords

pathology image analysis; convolutional neural network; unsupervised learning; semi-supervised learning

1. Introduction

It is widely accepted that understanding and curing complex diseases require systematic examination of disease mechanisms at multiple biological scales and integration of information from multiple data modalities [1, 2, 3, 4]. Tissue specimens have long been used to examine how disease manifests itself at the sub-cellular level and modifies tissue morphology [5, 6]. Advances in digital pathology imaging have made it feasible to capture high-resolution tissue images rapidly and facilitated quantitative analyses of tissue image data. Nuclear characteristics, such as size, shape and chromatin pattern, are important factors in distinguishing different types of cells and diagio nosing disease. Automatic analysis of nuclei can provide quantitative measures and new insights to disease mechanisms that cannot be gleaned from manual, qualitative evaluations of tissue specimens. Nucleus/cell detection and segmentation is a common methodology in histopathology image analysis [7, 8, 9] - a recent survey of nucleus segmentation algorithms can be found in [8]. Xie et al. and Xing et al. [10, 11] show i5 that it is difficult to develop highly accurate, robust and efficient tissue image segmentation algorithms. The algorithmic issues arise from heterogeneity in structure and texture characteristics across tissue specimens from different disease regions and from different disease subtypes. Even within a tissue specimen, structure and texture characteristics can vary from region to region. Moreover, images may contain tissue prepa- 20 ration, staining and image acquisition artifacts, such as variation in staining intensity and folded tissue regions, which create problems for computer algorithms.

In this work, we research and evaluate deep learning methods, more specifically Convolutional Neural Networks [12, 13] - a recent survey of research on CNNs can be found in [13], for detection and segmentation of nuclei in 20X histopathology image 25 patches with a typical resolution of 50 by 50 square microns (100 by 100 pixels). We design a novel Convolutional Autoencoder (CAE) that unifies nuclei detection and feature/representation learning in a single network and can be trained end-to-end without supervision. We also show that with existing labeled data, our network can be easily fine-tuned with supervision to improve the state-of-the-art performance of nuclei 30 classification and segmentation. Our contributions can be summarized as follows.

First, we propose a CAE architecture with crosswise sparsity that can *detect* and represent nuclei in histopathology images with the following advantages: A primary contribution of our work is the development of an unsupervised detection network by using the characteristics of histopathology image patches. To the best of our knowl- edge, this is the first unsupervised detection network in this type of computer vision application. Our method can be fine-tuned for end-to-end supervised learning. **Second,** our experimental evaluation using multiple datasets shows the proposed approach performs significantly better than other methods. The crosswise constraint in our method boosts the performance substantially. Our

unsupervised CAE achieves comparable nu- cleus detection results compared to recent supervised nucleus detection methods. (b) Our method achieves comparable results with 5% of training data needed by other methods, resulting in considerable savings in the cost of label generation. Our method reduces the error of the U-net method [14] by 20% in nucleus segmentation. **Third,** we eliminate pooling layers and strided convolutional layers in the CAE and CNN architec- ture for the nucleus segmentation task. Since each nucleus has only around 400 pixels, such spatial reduction layers discard important spatial information for pixel-wise segmentation. Our approach outputs more accurate segmentation predictions. The CAE architecture presented here has been employed as one component of a complex CNN based architecture employed in a National Cancer Institute sponsored cancer research project - the Pan Cancer Research Atlas. This cancer research project involved characterization of tumor infiltrating lymphocyte patterns in whole slide images along with elucidation of relationships between lymphocyte patterns and molecular tumor characterization [15]. The characterization analysis involved roughly 5,000 whole slide images obtained from 13 cancer types.

2. Related Work

Image segmentation is a fundamental image analysis method in computer vision [16, 17, 8, 18]. Machine learning has been extensively employed in image analysis tasks in biomedical research [19, 7, 20, 21, 22]. Rasti et al. [23] proposed an approach that employs a mixture ensemble of CNNs (ME-CNNs) to tumor classification in DCE-MRI breast cancer images. Their approach shows good performance even when trained with limited number of image samples. A multi-crop CNN method is proposed by Shen et al. [24] for classification of lung module malignancy in CT images. Their method models salient information in images through a multi-crop pooling strategy, which extracts regions from convolutional feature maps and executes max-pooling different times. Manivannan et al. [25] developed a method to detect and classify subcellular patterns in images of HEp-2 cells. The method trains ensembles of SVMs to classify cells into multiple classes. Tajbakhsh and Suzuki [26] compared massive-training artificial neural networks (MTANNs) and convolutional neural networks (CNNs) for lung nodule detection and classification. Khatami et al. [27] investigated the use of manifold learning for classification, regression and hypothesis testing with diffusion MR images. Their results show improved accuracy for supervised classification and regression and increased power for hypothesis testing.

Development of methods for tissue image analysis remains an active area of research. Numerous studies have devised methods for the detection, extraction, recog- nition of pathological patterns at various scales [7, 8, 28, 29]. Zheng et al. [28] developed a method based on convolutional neural networks (CNNs) to extract features from histopathology images, including patterns and distributions of nuclei. Al-Milaji et al. [29] proposed a CNN method for identification of stromal and epithelial tissue regions in images obtained from H&E stained tissue specimens. Deep learning based so automatic nuclei analysis methods [30, 31, 32, 33, 34, 35] requires a large-scale annotated dataset. Collecting annotated data is a labor intensive and challenging process since it requires the involvement of expert pathologists whose time is a very limited and expensive resource [36]. Thus many state-of-the-art nucleus analysis methods are semi-supervised [37, 38, 39, 40, 41]. They pretrain an

autoencoder for unsupervised representation learning and construct a CNN from the pretrained autoencoder. To better capture the visual variance of nuclei, one usually trains the unsupervised autoencoder on image patches with nuclei in the center [42,43,44]. This requires a separate nucleus detection step [34] which in most cases needs tuning to optimize the final classification performance. Instead of tuning the detection and classification modules separately, recent works [45, 46, 47, 48] successfully trained end-to-end CNNs to perform these tasks in an unified pipeline. Prior work has developed and employed supervised networks. Unsupervised detection networks do not exist in any visual application domains, despite the success of unsupervised learning in other tasks [49, 50].

3. Overview of Our Method

We design a novel Convolutional Autoencoder (CAE) that unifies nuclei detection and feature/representation learning in a single network and can be trained end-to-end without supervision. Our approach modifies the conventional CAE to encode not only appearance, but also spatial information in feature maps. To this end, our CAE first learns to separate background (eg.cytoplasm) and foreground (eg.nuclei) in an image patch, as shown in Fig. 2. We should note that an image patch is a rectangular region in a whole slide tissue image. We use image patches, because a tissue image can be very large and may not fit in memory. It is common in tissue image analysis to partition tissue images into patches and process the patches. We will refer to the partitioned image patches simply as the images. The CAE encodes the input image in a set of low resolution feature maps (background feature maps) with a small number of encoding neurons. The feature maps can only encode large scale color and texture variations because of their limited capacity. Thus these feature maps encode the image background. The high frequency residual between the input image and the reconstructed background is the foreground that contains nuclei.

We design our network to learn the foreground feature maps in a “crosswise sparse” manner: neurons across all feature maps are not activated (output zero) in most feature map locations. Only neurons in a few feature map locations can be activated. Since the non-activated neurons have no influence in the later decoding layers, the image foreground is reconstructed using only the non-zero responses in the foreground encoding feature maps. This means that the image reconstruction error will be minimized only if the activated encoding neurons at different locations capture the detected nuclei.

Learning a set of crosswise sparse foreground encoding feature maps is not straightforward. Neurons at the same location across all foreground feature maps should be synchronized: they should be activated or not activated at the same time depending on the presence of nuclei. In order to achieve this synchronization, the CAE needs to learn the locations of nuclei by optimizing the reconstruction error. Hence, the nucleus detection and feature extraction models are learned simultaneously during optimization. To represent the inferred nuclear locations, we introduce a special binary feature map: the nucleus detection map. We make this map sparse by thresholding neural activations. After optimization, a neuron in the nucleus detection map should output 1, if and only if there is a detected nucleus at the neuron’s location. The foreground feature maps are computed by element-wise multiplications between the nucleus detection map and a set of dense feature maps (Fig. 2).

In the next section we first introduce the CAE then describe our crosswise sparse CAE in detail.

4. Crosswise Sparse CAE

4.1. CAE for Semi-supervised CNN

An autoencoder is an unsupervised neural network that learns to reconstruct its input. The main purpose of this model is to learn a compact representation of the input as a set of neural responses [51]. A typical feedforward autoencoder is composed of an encoder and a decoder, which are separate layers. The encoding layer models the appearance information of the input. The decoder reconstructs the input from neural responses in the encoding layer. The CAE [38] and sparse CAE [37,40,52] are autoencoder variants. One can construct a CNN with a trained CAE. Such semi-supervised CNNs outperform fully supervised CNNs significantly in many applications [53, 42].

The architecture of our CAE is shown in Fig. 2. We train the CAE to minimize the input image reconstruction error. The early stages of the CAE network consists of six convolutional and two average-pooling layers. The network then splits into three branches: the nucleus detection branch, the foreground feature branch, and the background branch. The detection branch merges into the foreground feature branch to generate the foreground feature maps that represent nuclei. The foreground and background feature maps are decoded to generate the foreground and background reconstructed images. Finally the two intermediate images are summed to form the final reconstructed image.

4.2. Background Encoding Feature Maps

We first model the background (tissue, cytoplasm etc.) then extract the foreground that contains nuclei. Usually a majority of a tissue image will be background. The texture and color of the background vary usually in a larger scale compared to the foreground. Thus, a few small dense feature maps capture background information, because parts of the image encoded by these feature maps have the two properties that match the background: these parts are distributed throughout the whole image (because these feature maps are *dense*) and a larger scale texture and color (because there are limited number of neurons in these feature maps). In practice we represent the background of a 100×100 image by five 5×5 maps.

Large but crosswise sparse feature maps (foreground encoding maps) can only re- construct color and texture at *sparse* locations. However, the background has large- scale color and texture through the *whole* patch. By minimizing the reconstruction error, a few small dense feature maps must encode the background information which cannot be encoded by the crosswise sparse feature maps.

4.3. Foreground Encoding Feature Maps

Once the background is encoded and then reconstructed, the residual between the reconstructed background and the input image will be the foreground. The foreground consists of nuclei which are roughly of the same scale and often disperse throughout the image. The foreground encoding feature maps encode everything about the nuclei, including

their locations and appearance. A foreground feature map can be viewed as a matrix, in which each entry is a vector (a set of neuron responses) that encodes an image patch (the neurons' receptive field). The vectors will encode nuclei, if there are nuclei at the center of the image patches. Otherwise the vectors contain zeros only. Since a small number of non-zero vectors encode nuclei, the foreground feature map will be sparse.

4.3.1. Crosswise Sparsity—We formally define crosswise sparsity as follows: We denote a set of f feature maps as $X_1, X_2, \ldots X_f$. Each feature map is a matrix. We denote the i, j-th entry of the l-th feature map as $X_l^{i,j}$, and the size of a feature map is s × s. A conventional sparsity constraint requires:

$$\frac{\sum_{i,j,l} 1\left(X_l^{i,j} \neq 0\right)}{fs^2} \ll 1 \quad (1)$$

where 1(•) is the indicator function that returns 1 if its input is true and 0 otherwise. The crosswise sparsity requires:

$$\frac{\sum_{i,j} 1\left(\sum_l 1\left(X_l^{i,j} \neq 0\right) > 0\right)}{s^2} \ll 1 \quad (2)$$

In other words, in most locations in the foreground feature maps, neurons across *all* the feature maps should *not* be activated. This sparsity definition, illustrated in Fig. 3, can be viewed as a special form of group sparsity [54, 55].

If a foreground image is reconstructed by feature maps that are crosswise sparse, iso as defined by Eq. 2, the foreground image is essentially reconstructed by a few vectors in the feature maps. As a result, those vectors must represent salient objects in the foreground image- nuclei, since the CAE aims to minimize the reconstruction error.

4.3.2. Ensuring Crosswise Sparsity—Crosswise sparsity defined by Eq. 2 is not achievable using conventional sparsifi- cation methods [52] that can only satisfy Eq. 1. We introduce a binary matrix D with its i, j-th entry $D^{i,j}$ indicating if $X_l^{i,j}$ are activated for any l or not:

$$D^{i,j} = 1\left(\sum_l 1\left(X_l^{i,j} \neq 0\right) > 0\right) \quad (3)$$

Therefore Eq. 2 becomes:

$$\frac{\sum_{i,j} D^{i,j}}{s^2} \ll 1 \quad 4$$

The foreground feature maps X_1; X_2, … X_f are crosswise sparse, *iff* there exists a matrix D that satisfies Eq. 3 and Eq. 4. To satisfy Eq. 3, we design the CAE to generate a binary sparse feature map that represents D. The CAE computes X_l based on a dense feature map X'_l and D by element-wise multiplication:

$$X_l = X'_l \odot D \quad 5$$

We call the feature map D the detection map, shown in Fig. 2. The dense feature map X'_l, is automatically learned by the CAE by minimizing the reconstruction error.

The proposed CAE also computes the D that satisfies Eq. 4. Notice that Eq. 4 is equivalent to the conventional sparsity defined by Eq. 1, when the total number of feature maps $f = 1$ and X_f is a binary feature map. Therefore, Eq. 4 can be satisfied by existing sparsification methods. A standard sparsification methods is to add a sparsity penalty term in the loss function [52]. This method requires parameter tuning to achieve the desired expected sparsity. The k-sparse method [56] guarantees that exactly k neurons will be activated in D, where k is a predefined constant. However, in tissue images, the number of nuclei per image varies; the sparsity rate also should vary.

In this paper, we propose to use a threshold based method that guarantees an overall expected predefined sparsity rate, even though the sparsity rate for each CAE's input can vary. We compute the binary sparse feature map D as output from an automatically learned input dense feature map D':

$$D^{i,j} = sig\left(r\left(D'^{i,j} - t\right)\right) \quad 6$$

where sig(·) is the sigmoid function, r is a predefined slope, and t is an automatically computed threshold. We choose $r = 20$ in all experiments, making D a binary matrix in practice. Different r values do not affect the performance significantly based on our experience. Our CAE computes a large t in the training phase, which results in a sparse D. We define the expected sparsity rate as $p\%$, which can be set according to the average number of nuclei per image. We determine the sparsity rate, $p\%$, by randomly sampling 20 unlabeled lung adenocarcinoma patches and counting the average number of nuclei per patch. We then compute the desired p ($p = 1.6$ in all experiments) such that the number of activated neurons in the detection map equals to the count of average number of nuclei in a patch. This process takes less than 20 minutes. We compute t as

$$t = \mathrm{E}[\mathrm{percentile}_p(D'^{i,j})], \quad 7$$

where $\mathrm{percentile}_p(D'^{i,j})$ is the p-th percentile of $D'^{i,j}$ for all i,j, given a particular CAE's input image. In the training phase, we compute t using the running average method: $t \leftarrow (1 - \alpha)t + \alpha\ \mathrm{percentile}_p(D'^{i,j})$. We set the constant $\alpha = 0.1$ in all experiments. This running average approach is also used by batch normalization [57]. To make sure the running average of t converges, we also use batch normalization on $D'^{i,j}$ to normalize the distribution of $D'^{i,j}$ in each stochastic gradient descent batch.

In total, three parameters are introduced in our CAE: r, p, and α. The sparsity rate p can be decided based on the dataset easily. The other two parameters do not affect the performance significantly in our experiments. After the training phase, the threshold t is fixed as a constant.

With crosswise sparsity, each vector in the foreground feature maps can possibly 205 encode multiple nuclei. To achieve one-on-one correspondence between nuclei and encoded vectors, we simply reduce the size of the encoding neurons' receptive fields, such that a vector encodes a small region that is in the same size of a nucleus.

5. Experiments

We initialize the parameters of CNNs with the parameters of our trained cross- wise sparse CAEs. We empirically evaluate this approach on four datasets: a selfcollected lymphocyte classification dataset, the nuclear shape and attribute classification dataset [43], the CRCHistoPhenotypes nucleus detection dataset [34], and the MICCAI 2015 nucleus segmentation challenge dataset [58].

5.1. Datasets

Dataset for Unsupervised Learning.—We collected 0.5 million unlabeled small images randomly cropped from 400 lung adenocarcinoma histopathology images obtained from the public TCGA repository [60]. The cropped images are 100×100 pixels in 20X (0.5 microns per pixel). We will refer to cropped images simply as images in the rest of this section.

Datasets for Nucleus Classification (Sec. 5.6.1).—We evaluate the classification performance of our method on two datasets: a self-collected lymphocyte classification dataset, and the nuclear shape and attribute classification dataset [43].

Lymphocyte is a type of white blood cell in the immune system. Automatic recognition of lymphocytes is very important in many situations including the study of can- cer immunotherapy [61, 62, 63]. We collected a dataset of 1785 images of nuclei that were labeled lymphocyte or non-lymphocyte by a pathologist. These 1785 images were cropped from 12 representative lung adenocarcinoma whole slide tissue images from the TCGA repository [60]. We use labeled images cropped from 10 whole slide tissue images as the

training set and the rest as the test set. We show randomly selected image 230 examples in Fig. 5. In addition, we apply our method on an existing dataset [43] for nuclear shape and attribute classification. The dataset consists of 2000 images of nuclei labeled with fifteen morphological attributes and shapes.

Dataset for Nucleus Detection Experiments (Sec. 5.6.2).—We test our method for nucleus detection using the CRCHistoPhenotypes nucleus detection dataset [34] which 235 contains 100 colorectal adenocarcinoma images of 500×500 pixels. In total there are 29,756 marked locations of nuclei.

Dataset for Nucleus Segmentation Experiments (Sec. 5.6.4).—For training, we use the MICCAI 2015 nucleus segmentation challenge dataset [58] which contains 15 training images. The ground truth masks of nuclei are provided in the training dataset.

In addition, we collect a large-scale weakly supervised nucleus segmentation training set with DAPI staining techniques. It contains 763 images of 500×500 pixels. For testing, we use the MICCAI 2015 nucleus segmentation challenge test set which contains 18 images. A typical resolution of the MICCAI 2015 images is 500×500.

5.2. CAE Architectures

Our CAEs in all experiments are trained on the same unlabeled dataset, the 0.5 million lung adenocarcinoma image patches. The CAE architectures are different depending on applications.

CAEs for Classification and Detection.—We use the same architecture illustrated in Fig. 2, Tab. 1 and Tab. 2. Note that we apply batch normalization [57] before the 250 leaky ReLU activation function [59] in all layers.

CAEs for Nucleus Segmentation.—The average size of nuclei in the dataset for nucleus segmentation experiments (Sec. 5.6.4) is around 20×20 pixels. Therefore pooling and strided convolutional layers can discard important spatial information which is important for pixel-wise segmentation. The U-net [14] addresses this issue by adding skip connections. However, we find in practice that eliminating pooling and strided convolutional layers completely yields better performance. The computational complexity is very high for a network without any spatial reduction layers. Thus, compared to Tab. 1 and Tab. 2, we use smaller input dimensions (40 × 40) and fewer (80 to 200) feature maps in the CAE for nucleus segmentation. Other settings of the CAE for segmentation remain unchanged.

5.3. CNN Architectures

We construct all of our supervised CNNs based on trained CAEs. We use CAE trained on lung adenocarcinoma patches to initialize the CNNs. Note that the proposed CAE is fully convolutional and can initialize CNNs with different size inputs.

For the classification tasks (Sec. 5.6.1), the supervised CNN is constructed from Parts 1–6 of the CAE. We initialize the parameters in these layers to be the same as the parameters in the CAE. We attach four 1 × 1 convolutional layers after the foreground encoding layer and two

3×3 convolutional layers after the background encoding layer. Each added layer has 320 convolutional filters. We then apply global average pooling 270 on the two branches. The pooled features are then concatenated together, followed by a final classification layer with sigmoid activation function.

For the nucleus detection task (Sec. 5.6.2), the supervised CNN is constructed from Parts 1, 2, 5 of the CAE. After Part 5, we attach five 1 × 1 convolutional layers. The activation function of the last layer is sigmoid. Thus, our detection CNN outputs a 275 probability map (matrix). The ground truth for training the detection CNN is a binary matrix with the same size of the output probability map. We do not model the problem of touching cells explicitly. The CAE that initializes the detection CNN is forced to represent a large clump of nuclei by multiple separate detected objects. The receptive field of one encoding neuron in the CAE is designed to be large enough to contain only 280 one regular size nucleus in most cases. Note that the size of the output probability map is one quarter of the size of the input image. In order to obtain pixel-level nucleus detection results, after obtaining the predicted probability map, we resize it with bilinear interpolation to the same size of the input image.

For the nucleus segmentation task (Sec. 5.6.4), the supervised CNN is constructed from Parts 1 and 3 of the CAE which forces the segmentation CNN to learn separate representations for each nucleus. The final segmentation of each nucleus is computed from the separate intermediate representations. The training label is class-level (not instance-level) for all segmentation CNNs. Currently we do not model the touching cell/nuclei problem explicitly. Each segmented region is considered one nucleus. After Part 3, we add six 3 × 3 convolutional layers followed by a segmentation layer. The segmentation layer is the same to the patch-CNN's [64] segmentation layer which is a fully-connected layer with sigmoid activation function followed by reshaping. In addition, inspired by the U-net [14], we add two skip connections in the network. Note that the skip connections are added when constructing the CNN only, not on the CAE.

For all tasks, we randomly initialize the parameters of the added layers. We train the parameters of the added layers until convergence before fine-tuning the whole network.

5.4. Training and Testing Details

We train our CAE on the unlabeled dataset, minimizing the pixel-wise root mean squared error between the input images and the reconstructed images. We use stochas- tic gradient descent with batch size 32, learning rate 0.03 and momentum 0.9. The loss converges after 6 epochs. We show randomly selected examples of the nucleus detection feature map as well as the reconstructed foreground and background images in Fig. 4. The crosswise sparsity does not guarantee that the foreground pixels get activated if there is a nucleus. As is shown in Fig. 4, however, during optimization of the 305 reconstruction loss, the foreground encoding feature maps detect and encode the position and appearance of nuclei. This is because the background encoding feature maps can only encode large scale color information (thus foreground pixels) and are responsible for reconstructing the details of the input patch such as nuclei. The performance of unsupervised detection of nuclei is reported in Tab. 4.

For the CNN (constructed from the CAE) training, we use stochastic gradient descent with batch size, learning rate, and momentum selected for each task independently. For all tasks, we divide the learning rate by 10 when the error has plateaued. We use sigmoid as the nonlinearity function in the last layer and log-likelihood as the loss function. We apply three types of data augmentation. First, the input images are 315 randomly cropped from a larger image. Second, the colors of the input images are randomly perturbed. Third, we randomly rotate and mirror the input images. During testing, we average the predictions of 25 image crops. We implemented our CAE and CNN using Theano [65]. We trained the CAE and CNN on a single Tesla K40 GPU (1/4 of the speed of GTX 1080TI).

The proposed CAE takes less than 0.01 seconds on a Tesla K40 GPU to compute the encoding feature maps for one input patch in the test phase using the network in Tab. 1. For a typical size slide which contains 125k non-overlapping tissue patches, thepro- posed networks take 21, 19 and 18 minutes, respectively, for all the patches. To speed up the testing phase, we plan to reimplement the network (currently running under Theano 0.9) using an updated deep learning toolbox, which supports more recent versions of CUDA and cuDNN. With recent deep learning hardware such as Nvidia V100, we expect the testing time to drop significantly per WSI. We will also consider incorporating recent techniques for reducing the network size such as the SqueezeNet [66].

5.5. Methods Tested

CSP-CAE Our crosswise sparse CAE shown in Fig. 2. We use the detection map directly as an unsupervised nucleus detection output. This method is only evaluated on the nucleus detection dataset.

CSP-CNN Our CNN initialized by the proposed crosswise sparse CAE shown in Fig. 2. The CNN construction is described in Sec. 5.3. We set the sparsity rate to 1.6%, such that the number of activated foreground feature map locations roughly equals to the average number of nuclei per image in the unsupervised training set.

SUP-CNN A fully supervised CNN. Its architecture is similar to our CSP-CNN except that: 1). There is no background representation branch (no Part 4, 8 in Fig. 2). 2). There is no nucleus detection branch (no Part 2, 5 in Fig. 2). The SUP-CNN has a very standard architecture, at the same time similar to our CSP-CNN.

SUP-CSP-CNN A fully supervised CNN with the exact same architecture as the proposed CSP-CNN. It is trained from random initialization instead of an unsupervised CSP-CAE.

U-NET We use the authors' U-net architecture and implementation [14] for nucleus segmentation and detection. The U-net is fully supervised and not initialized by an autoencoder. We test five U-nets with the same architecture but different number of feature maps per layer and select the best performing network. All five U-nets perform similarly.

DEN-CNN CNN initialized by a conventional Convolutional Autoencoder (CAE) with- 350 out the sparsity constraint. Its architecture is similar to our CSP-CNN except that it has no

nucleus detection branch. In particular, there is no Part 2 and Part 5 in Fig. 2 and Part 6 is an identity mapping layer.

SP-CNN CNN initialized by a sparse CAE without the crosswise constraint. Its architecture is similar to our CSP-CNN except that it has no nucleus detection branch 355 and uses the conventional sparsity constraint defined by Eq. 1. In particular, there is no Part 2 and Part 5 in Fig. 2 and Part 6 is a thresholding layer: define its input as D', its output $D = \mathrm{ReLU}(D' - t)$, where t is obtained in the same way defined by Eq. 7. We set the sparsity rate to 1.6% which equals to the rate we use in CSP-CNN.

VGG16 We fine-tune the VGG 16-layer network [67] which is pretrained on Ima- 360 geNet [68]. Fine-tuning the VGG16 network has been shown to be robust for pathology image classification [69,42].

5.6. Results

5.6.1. Classification of Nuclei—We evaluated our method on two nucleus classification datasets. The first dataset is used to classify lymphocytes vs. non-lymphocytes. We compared our method with an unsupervised nucleus detection and feature extraction method [70], which is based on level sets. We split training and testing images 4 times and average the results. As the baseline method we carefully tuned the unsupervised method [70] and applied a multilayer neural network on top of the extracted features. We should note that the feature extraction step and the classification step have to be tuned separately in the baseline method, whereas our CSP-CNN method can be trained end-to-end. As is shown in Tab. 3, CSP-CNN achieves the best results and reduces the error of SP-CNN by **25%.**

The second dataset is the nuclear shape and attribute classification dataset [43]. For this task, we adopt the same 5-fold training and testing data separation protocol and report the results in Tab. 3. Our methods improves less over the state-of-the-art with this dataset than with the other datasets, because the images of nuclei are results of a fixed nucleus detection method which we cannot fine-tune with our proposed method.

5.6.2. Detection of Nuclei—We use a sliding window approach to train and test our methods. A CNN outputs a 380 feature map of one quarter the size of its input. The output map is resized with bilinear interpolation to the same size of the input image. Finally we apply Gaussian filtering followed by non-maximum suppression and thresholding to obtain detected nucleus locations. For evaluation, we follow the standard 2-fold cross-validation method used in the baseline method [34]. A detected nucleus location is correct if there is a ground 385 truth nucleus location within 6 pixels. If there are multiple detected locations within 6 pixels of a ground truth location, only the nearest detected location is considered correct and all other detections are considered false positives. We achieve state-of- the-art results with this dataset (see Tab. 4). Even with no supervision, our crosswise sparse CAE (CSP-CAE) trained on lung adenocarcinoma image patches outperforms supervised methods trained on the CRCHistoPhenotypes colorectal adenocarcinoma dataset. We show randomly selected detection results in Fig. 6.

5.6.3. Evaluation of unsupervised nucleus detection and representation—We show the performance of the proposed CAE on unsupervised detection and feature extraction of nuclei with the CRCHistoPhenotypes nucleus detection dataset [34] and our lymphocyte classification dataset. Additionally, we show the effect of the sparsity rate p for both tasks. Recall that we determine the sparsity rate, $p\%$ by the process described in Section 4.3.2 by randomly sampling 20 unlabeled lung adenocarcinoma (LUAD). To assess the variability of p, we repeat the process three times and have $p = 1.664$, $p = 1.696$, $p = 1.720$. In our experiments, we use $p = 1.6$. In this section, we also test $p = 1.2$ and $p = 2.0$ with the proposed CAE.

We show experimental results for our unsupervised detection method CSP-CAE in Tab. 5. To evaluate the unsupervised nucleus representation features, we use the features with a Multi-Layer Perceptron for lymphocyte classification. We name the method CSP-CAE-MLP and show performance results in Tab. 6. To implement the CSP-CAE-MLP, we simply use the CSP-CNN but fix all its parameters initialized by the CSP-CAE (we only train the multi-layer perceptron on top of the representation features). In both experiments, our method achieves comparable results to many of the existing supervised methods. We should also note that $p = 1.2$ and $p = 1.6$ yield similar results as $p = 1.6$ which is used in all of the other experiments.

5.6.4. Segmentation of Nuclei—We use a sliding window approach to train and test our CNNs. A CNN outputs a feature map of the same size as its input. For evaluation, we follow the standard metric used in the MICCAI challenge: the DICE-average (average of two different versions of the DICE coefficient). We show results in Tab. 7. The proposed method achieves a significantly higher score than that of the challenge winner [33] and U-net [14]. Because the size of nuclei are only around 20 × 20 pixels, we eliminate our network's pooling layers completely and use no strided convolutional layers to preserve spatial information. We believe this is an important reason our method outperforms U- net. We show randomly selected segmentation examples in Fig. 7.

5.6.5. Training CNN with Weakly Labeled Data—Manual generation of training datasets for segmentation is a labor intensive and time consuming process. Even a relatively small patch in a tissue image can contain hundreds or thousands of nuclei. Manual segmentation of nuclei in such patches can take several hours. For example, the preparation of the MICCAI challenge dataset took several weeks. Multiple students were hired to manually segment each and every nucleus in a set of patches. Work done by each student was reviewed by pathologists to refine the segmentations and produce accurate results. This process generates highly accurate training data, which we call *strongly labeled data.*

In some studies, multiple types of staining are applied on tissue specimens. For example, a tissue slice may be stained with the Hematoxylin and Eosin (H&E) stain and imaged. The same tissue slice may then be rinsed to remove the H&E stain and re-stained with the immunohistochemistry (IHC) or DAPI stain and imaged. We have examined the utility of images from DAPI stained tissue specimens to produce training segmentation datasets for CNN. Tissue images from DAPI stained tissue specimens exhibit higher contrast between background and nuclei. We used this characteristic of the DAPI images to generate

segmentation masks using a parameterized segmentation algorithm. We call this type of training data *weakly labeled data*. Fig. 8 shows two randomly selected examples of DAPI stained images with corresponding H&E images.

In our experiment, H&E slides were first digitized under 20X objective with Olym- pus VS120 whole slide scanner, then de-coverslipped in Acetone and rinsed in 100% Alcohol, as well as descending percentages. De-staining was carried out by sequentially rinsing slides in deionized water, 1% potassium permanganate (1 min), water, and 2% Oxalic Acid (30 sec or as long as it takes to bleach out potassium permanganate). Finally the slides were rinsed in water before coverslipped with DAPI (hardset) using #1 coverslips. The DAPI re-stained slides were ringed with nail polish to seal and once again imaged with VS120. The DAPI images and the corresponding H&E images were obtained from a tissue microarray (TMA). This microarray contained 100 disc images. Each disc was originated from a separate tissue specimen. The respective DAPI and H&E disc images were registered using an FFT-based registration method [72]. We employed a level-set based segmentation algorithm [73] to segment the DAPI images. The algorithm first converts H&E images to gray scale for segmentation. The DAPI images were already converted to gray scale during the image acquisition and postimaging steps. After the DAPI images had been segmented, the output masks were overlayed on the matching H&E images. The overlaid images were then reviewed by a pathologist who selected a subset of the images for inclusion in the training dataset.

The CNN was trained with using the selected masks generated from the DAPI images along with the respective H&E images. The trained CNN was applied on the MICCAI 2015 segmentation challenge test set in the test phase. We computed average DICE coefficient values for the CNN models trained using the DAPI training dataset, the MICCAI 2015 training dataset, and a dataset containing both the DAPI and MIC- CAI training datasets. Training using DAPI images shows good results. The CNN trained with the DAPI images achieved a DICE coefficient of 0.77. The DICE coefficient value of the CNN trained with the MICCAI 2015 training dataset was higher at 0.87. We attribute this to the fact that the MICCAI training dataset is generated through a meticulous, yet very time-consuming, manual process and have much more accurate nucleus boundaries. Inaccuracies in the DAPI based training dataset stems from: (1) DAPI stained cells, though remained firmly in place, bear subtle morphological changes from H&E due to additional chemical treatment applied, this resulted in the fact that registration between a DAPI image and an H&E image is not perfect. Hence, segmentation boundaries from the DAPI image will not match the actual boundaries of nuclei in the H&E image. (2) Boundaries generated from computer segmentation algorithms generally are not as accurate and tight as manual segmentations. Nevertheless, a primary advantage of using DAPI images is that we are able to generate the training dataset in a few days compared with multiple weeks for the MICCAI challenge dataset.

6. Conclusions

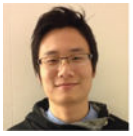
We propose a crosswise sparse CAE that uses the visual characteristics of nuclei for unsupervised nucleus detection and feature extraction simultaneously. Using the CAE to

initialize a supervised CNN makes it possible to carry out the nucleus detection, feature extraction, and classification/segmentation training steps in an end-to-end fash-ion. Our experimental evaluation shows that this approach performs much better than other approaches and that the crosswise constraint plays an important role in boosting performance. In addition, our approach achieves comparable results with 5% of training data needed by other methods. We also investigated the use of weakly labeled data generated from DAPI stained images for training. An experimental evaluation showed this approach achieves good results. Generating ground truth data in digital pathology is a labor-intensive process. This can be a limiting factor in the application of deep learning methods. The use of crosswise sparse CAE and weakly labeled data addresses this problem and can lead to more effective application of deep learning in digital pathology. In future work, we plan to use domain knowledge to regularize the encoding layers, using techniques such as the N-cut loss [74] and better detect nuclei of various shapes and texture. For the supervised instance-level segmentation of nuclei, we will test the deep watershed method. To speed up the testing phase, we will investigate techniques for reducing the network size such as the SqueezeNet [66].

Acknowledgments.

This work was supported in part by 1U24CA180924–01A1 from the National Cancer Institute, R01LM011119–01 and R01LM009239 from the National Library of Medicine, and a grant from Ecole CentraleSupelec Paris.

Author Biography



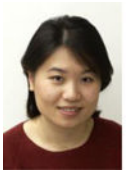
Le Hou is a PhD. candidate working with Prof. Dimitris Samaras and Prof. Joel Saltz at Stony Brook University. Previously, he worked as a senior software engineer at Baidu INC. He graduated as a Bachelor of Computer Science and Technology from Huazhong University of Science and Technology.

Vu Nguyen is currently a PhD. student under the supervision of Prof. Dimitris Samaras and Prof. Joel Saltz at Stony Brook University. He received his BSc. and MSc. in Computer Science at University of Information Technology, VNU-HCM.

Ari Kanevsky is an undergraduate student in the departments of Computer Science and PreMedicine at Stony Brook University. He is also a researcher at the Montreal Institute for Learning Algorithms, working under Dr. Yoshua Bengio on biologically-plausible applications in Deep Learning.

Tahsin M. Kurc is a Research Associate Professor in the Department of Biomedical Informatics at Stony Brook University. He received a PhD degree in computer science from Bilkent University in Turkey. His research focuses on distributed and parallel computing, Grid computing, and systems software for large-scale and data-intensive scientific applications.

Tianhao Zhao, MD is a surgical pathology fellow at Stony Brook Medicine Department of Pathology. Previously she completed her pathology residency at Wake Forest Medical Center in Winston Salem, NC and her pathology informatics fellowship at Stony Brook Medicine Department of Bioinformatics. She received her MD at UTSouthwest- ern Medical Center in Dallas, TX.



Dr. Wenjin Chen obtained a Ph.D. at the Joint Program of Molecular Biosciences from Rutgers University and the former University of Medicine and Dentistry of New Jersey, and serves as the Associate Director, Biomedical Imaging at Center for Biomedical Imaging & Informatics, Rutgers Cancer Institute of New Jersey. Her research interests include digital pathology, whole slide image analysis and clinical data warehouse.

Dr. David Foran is Professor of Pathology, Laboratory Medicine & Radiology and Chief of the Division of Medical Informatics at Rutgers-Robert Wood Johnson Medical School. He

also serves as Executive Director of Computational Imaging & Biomedical informatics; and Chief Informatics Officer at Rutgers Cancer Institute of New Jersey.

Dr. Joel H. Saltz is Professor and Chair of Biomedical Informatics at Stony Brook University. He is an MD-PhD in Computer Science and a Boarded Clinical Pathologist with training at Duke and Johns Hopkins. He is a pioneer in digital Pathology with a 20 year history of research beginning with the developing of the first virtual microscope system at Johns Hopkins in 1997.

References

[1]. Zhang Z, Xie Y, Xing F, McGough M, Yang L, Mdnet: A semantically and visually interpretable medical image diagnosis network, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp.6428–6436.

[2]. Zhang Z, Chen P, Sapkota M, Yang L, Tandemnet: Distilling knowledge from medical images using diagnostic reports as optional semantic references, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp.320–328.

[3]. Huang H, Tosun AB, Guo J, Chen C, Wang W, Ozolek JA, Rohde GK, Cancer diagnosis by nuclear morphometry using spatial information, Pattern Recognition Letters 42 (2014)115–121. [PubMed: 24910485]

[4]. Lu C, Mandal M, Automated analysis and diagnosis of skin melanoma on whole slide histopathological images, Pattern Recognition 48 (8) (2015)2738–2750.

[5]. Pantanowitz L, Valenstein PN, Evans AJ, Kaplan KJ, Pfeifer JD, Wilbur DC, Collins LC, Colgan TJ, Review of the current state of whole slide imaging in pathology, Journal of pathology informatics 2.

[6]. Kononen J, Bubendorf L, Kallionimeni A, Barlund M, Schraml P, Leighton S, Torhorst J, Mihatsch MJ, Sauter G, Kallionimeni O-P, Tissue microarrays for high-throughput molecular profiling of tumor specimens, Nature medicine 4 (7) (1998)844–847.

[7]. Xing F, Xie Y, Su H, Liu F, Yang L, Deep learning in microscopy image analysis: A survey, IEEE Transactions on Neural Networks and Learning Systems PP (99) (2017)1–19.

[8]. Xing F, Yang L, Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review, IEEE reviews in biomedical engineering 9 (2016)234–263. [PubMed: 26742143]

[9]. Zhang W, Li H, Automated segmentation of overlapped nuclei using concave point detection and segment grouping, Pattern Recognition 71 (2017)349–360.

[10]. Xie Y, Zhang Z, Sapkota M, Yang L, Spatial clockwork recurrent neural network for muscle perimysium segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016.

[11]. Xing F, Shi X, Zhang Z, Cai J, Xie Y, Yang L, Transfer shape modeling towards high-throughput microscopy image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016.

[12]. Schmidhuber J, Deep learning in neural networks: An overview, Neural Networks 61 (2015)85–117. [PubMed: 25462637]

[13]. Gu J, Wang Z, et al., Recent advances in convolutional neural networks, Pattern Recognition 77 (2018)354–377.

[14]. Ronneberger O, Fischer P, Brox T, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, 2015, pp.234–241.

[15]. <Saltz b/>J., Gupta R, Hou L, et al., Spatial organization and molecular correlation of tumor infiltrating lymphocytes using deep learning on pathology images, Cell Reports, accepted for publication, 2018.

[16]. Peng B, Zhang L, Zhang D, A survey of graph theoretical approaches to image segmentation, Pattern Recognition 46 (3) (2013)1020–1038.

[17]. Xian M, Zhang Y, Cheng H, Xu F, Zhang B, Ding J, Automatic breast ultra-sound image segmentation: A survey, Pattern Recognition 79 (2018)340–355.

[18]. Bibiloni P, Gonzlez-Hidalgo M, Massanet S, A survey on curvilinear object segmentation in multiple applications, Pattern Recognition 60 (2016)949–970.

[19]. Suzuki K, Zhou L, Wang Q, Machine learning in medical imaging, Pattern Recognition 63 (2017)465–467.

[20]. Sertel O, Kong J, Shimada H, Catalyurek U, Saltz JH, Gurcan MN, Computer-aided prognosis of neuroblastoma on whole-slide images: Classification of stromal development, Pattern recognition 42 (6) (2009)1093–1103. [PubMed: 20161324]

[21]. Kong J, Sertel O, Shimada H, Boyer K, Saltz J, Gurcan M, Computer-aided evaluation of neuroblastoma on whole-slide histology images: Classifying grade of neuroblastic differentiation, Pattern Recognition 42 (6) (2009)1080–1092. [PubMed: 28626265]

[22]. Smochina C, Manta V, Kropatsch W, Crypts detection in microscopic images using hierarchical structures, Pattern Recognition Letters 34 (8) (2013)934–941.

[23]. Rasti R, Teshnehlab M, Phung SL, Breast cancer diagnosis in dce-mri using mixture ensemble of convolutional neural networks, Pattern Recognition 72 (2017)381–390.

[24]. Shen W, Zhou M, Yang F, Yu D, Dong D, Yang C, Zang Y, Tian J, Multicrop convolutional neural networks for lung nodule malignancy suspiciousness classification, Pattern Recognition 61 (2017)663–673.

[25]. Manivannan S, Li W, Akbar S, Wang R, Zhang J, McKenna SJ, An au- tomated pattern recognition system for classifying indirect immunofluorescence images of hep-2 cells and specimens, Pattern Recognition 51 (2016)12–26.

[26]. Tajbakhsh N, Suzuki K, Comparing two classes of end-to-end machine-learning models in lung nodule detection and classification: Mtanns vs. cnns, Pattern Recognition 63 (2017)476–486.

[27]. Khatami M, Schmidt-Wilcke T, Sundgren PC, Abbasloo A, Schlkopf B, Schultz T, Bundlemap: Anatomically localized classification, regression, and hypothesis testing in diffusion mri, Pattern Recognition 63 (2017)593–600.

[28]. Zheng Y, Jiang Z, Xie F, Zhang H, Ma Y, Shi H, Zhao Y, Feature extraction from histopathological images based on nucleus-guided convolutional neural network for breast lesion classification, Pattern Recognition 71 (2017)14–25.

[29]. Al-Milaji Z, Ersoy I, Hafiane A, Palaniappan K, Bunyak F, Integrating segmentation with deep learning for enhanced classification of epithelial and stromal tissues in h&e images, Pattern Recognition Letters (2017)1–8.

[30]. Xie Y, Kong X, Xing F, Liu F, Su H, Yang L, Deep voting: A robust ap- 580 proach toward nucleus localization in microscopy images, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp.374–382.

[31]. Xie Y, Xing F, Kong X, Su H, Yang L, Beyond classification: structured regression for robust cell detection using convolutional neural network, in: Medical Image Computing and Computer-Assisted Intervention, 2015, pp.358–365.

[32]. Wang S, Yao J, Xu Z, Huang J, Subtype cell detection with an accelerated deep convolution neural network, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016, pp.640–648.

[33]. Chen H, Qi X, Yu L, Dou Q, Qin J, Heng P-A, Dcan: Deep contour-aware networks for object instance segmentation from histology images, Medical image analysis 36 (2017)135–146. [PubMed: 27898306]

[34]. Sirinukunwattana K, Raza A, Tsang Y-W, Snead D, Cree I, Rajpoot N, Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images, Medical imaging 35 (5) (2016)1196–1206. [PubMed: 26863654]

[35]. Xu Z, Huang J, Detecting 10,000 cells in one second, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016.

[36]. Pathologists' Hourly Wages, http://www1.salary.com/Physician-Pathology-hourly-wages.html.

[37]. Ranzato M, Poultney C, Chopra S, LeCun Y, Efficient learning of sparse rep-resentations with an energy-based model, in: Advances in Neural Information Processing Systems, 2006, pp.1137–1144.

[38]. Masci J, Meier U, Ciresan D Schmidhuber J, Stacked convolutional autoencoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks (ICANN), 2011, pp.52–59.

[39]. Bayramoglu N, Heikkilä J, Transfer learning for cell nuclei classification in histopathology images, in: European Conference on Computer Vision Workshops, 2016, pp.532–539.

[40]. Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J, Madabhushi A, Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images, IEEE transactions on medical imaging 35 (1) (2016)119–130. [PubMed: 26208307]

[41]. Su H, Xing F, Kong X, Xie Y, Zhang S, Yang L, Robust cell detection and segmentation in histopathological images using sparse reconstruction and stacked denoising autoencoders, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp.383–390.

[42]. Hou L, Singh K, Samaras D, Kurc TM, Gao Y, Seidman RJ, Saltz JH, Automatic histopathology image analysis with CNNs, in: New York Scientific Data Summit, 2016.

[43]. Murthy V, Hou L, Samaras D, Kurc TM, Saltz JH, Center-focusing multitask CNN with injected features for classification of glioma nuclear images, in:Winter Conference on Applications of Computer Vision (WACV), 2017.

[44]. Gragnaniello D, Sansone C, Verdoliva L, Cell image classification by a scale and rotation invariant dense local descriptor, Pattern Recognition Letters 82 (2016)72–78.

[45]. Graves A, Jaitly N, Towards end-to-end speech recognition with recurrent neural networks., in: International Conference on Machine Learning, 2014.

[46]. Ren S, He K, Girshick R, Sun J, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp.91–99.

[47]. Redmon J, Divvala S, Girshick R, Farhadi A, You only look once: Unified, 630 real-time object detection, in: Proceedings of the IEEE conference on compute vision and pattern recognition (CVPR), 2016, pp.779–788.

[48]. Kokkinos I, Ubernet: Training auniversal'convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp.6129–6138.

[49]. Radford A, Metz L, Chintala S, Unsupervised representation learning with deep convolutional generative adversarial networks, in: International Conference on Learning Representations, 2016.

[50]. Doersch C, Gupta A, Efros AA, Unsupervised visual representation learning by context prediction, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp.1422–1430.

[51]. Deng L, Seltzer ML, Yu D, Acero A, Mohamed A.-r., Hinton GE, Binary coding of speech spectrograms using a deep auto-encoder., in: Eleventh Annual Conference of the International Speech Communication Association, 2010.

[52]. Ng A, Sparse autoencoder, Tech. rep., Lecture notes, Stanford University (2011).

[53]. Johnson R, Zhang T, Semi-supervised convolutional neural networks for text categorization via region embedding, in: Advances in Neural Information Processing Systems (NIPS), 2015, pp. 919–927.

[54]. Murdock C, Li Z, Zhou H, Duerig T, Blockout: Dynamic model selection for 650 hierarchical deep networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2016, pp.2583–2591.

[55]. B. Graham, Spatially-sparse convolutional neural networks, arXiv preprint arXiv:1409.6070.

[56]. Makhzani A, Frey B, k-sparse autoencoders, in: International Conference on 655 Learning Representations (ICLR), 2014.

[57]. Ioffe S, Szegedy C, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning (ICML), 2015, pp.448–456.

[58]. MICCAI 2015 workshop and challenges in imaging and digital pathol- ogy: The computational brain tumor cluster of event, https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=20644646 (2015).

[59]. Maas AL, Hannun AY, Ng AY, Rectifier nonlinearities improve neural network acoustic models, in: International conference on machine learning (ICML), Vol. 30, 2013, p.3.

[60]. The Cancer Genome Atlas, https://cancergenome.nih.gov/.

[61]. Galon J, Costes A, et al., Type, density, and location of immune cells within human colorectal tumors predict clinical outcome, Science 313 (5795) (2006)1960–1964. [PubMed: 17008531]

[62]. Salgado R, Denkert C, et al., The evaluation of tumor-infiltrating lymphocytes (TILs) in breast cancer: recommendations by an international TILs working group 2014, Annals of oncology 26 (2) (2014)259–271. [PubMed: 25214542]

[63]. Turkki R, Linder N, Kovanen PE, Pellinen T, Lundin J, Antibody-supervised deep learning for quantification of tumor-infiltrating immune cells in hematoxylin and eosin stained breast cancer samples, Journal of Pathology Informatics 7.

[64]. Vicente TFY, Hou L, Yu C-P, Hoai M, Samaras D, Large-scale training of shadow detectors with noisily-annotated shadow examples, in: European Conference on Computer Vision (ECCV), 2016, pp.816–832.

[65]. Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, arXiv preprint abs/1605.02688.

[66]. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size, arXiv preprint arXiv: 1602.07360.

[67]. Simonyan K, Zisserman A, Very deep convolutional networks for large-scale 685 image recognition, in: International Conference on Learning Representations (ICLR), 2015.

[68]. Russakovsky O, Deng J, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015)211–252.

[69]. Xu Y, Jia Z, Ai Y, Zhang F, Lai M, Eric I, Chang C, Deep convolutional 690 activation features for large scale brain tumor histopathology image classification and segmentation, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp.947–951.

[70]. Zhou N, Yu X, Zhao T, et al., Evaluation of nucleus segmentation in digital pathology images through large scale image synthesis, in: Medical Imaging 2017: Digital Pathology, Vol. 10140, International Society for Optics and Photonics, 2017, p.101400K.

[71]. Yuan Y, Failmezger H, Rueda OM, et al., Quantitative image analysis of cellular heterogeneity in breast tumors complements genomic profiling, Science translational medicine 4 (157).

[72]. Reddy BS, Chatterji BN, An fft-based technique for translation, rotation, and scale-invariant image registration, IEEE transactions on image processing 5 (8) (1996)1266–1271. [PubMed: 18285214]

[73]. Gao Y, Ratner V, Zhu L, Diprima T, Kurc T, Tannenbaum A, Saltz J, Hierarchical nucleus segmentation in digital pathology images, in: Medical Imaging 2016: Digital Pathology, Vol. 9791, International Society for Optics and Photonics, 2016, p.979117.

[74]. X. Xia, B. Kulis, W-net: A deep model for fully unsupervised image segmentation, arXiv preprint arXiv:1711.08506.

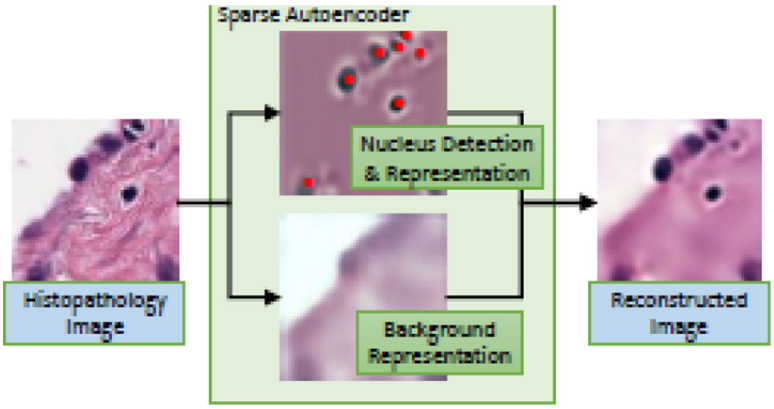


Figure 1:

Our autoencoder decomposes histopathology image patches and detect nuclei in a fully unsupervised fashion. It first decomposes an input image patch into foreground (eg.nuclei) and background (eg.cytoplasm). It then detects nuclei in the foreground by representing the locations of nuclei as a sparse feature map. Finally, it encodes each nucleus to a feature vector. Our autoencoder is trained end-to-end, minimizing the reconstruction error.

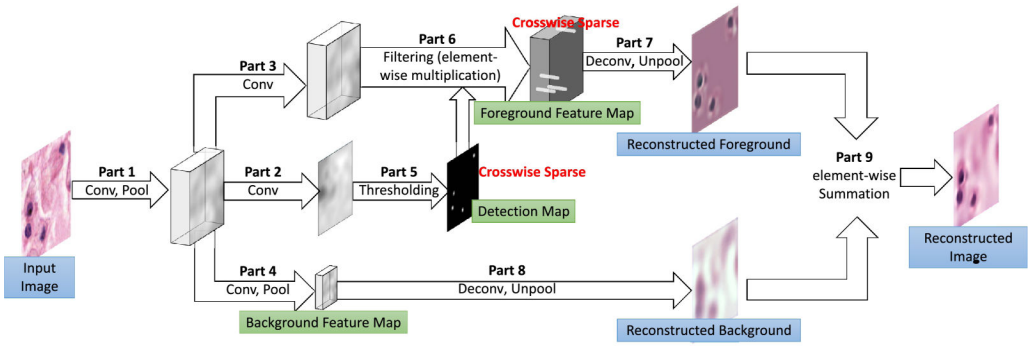


Figure 2:

The architecture of our sparse Convolutional Autoencoder (CAE). The CAE minimizes image reconstruction error. The reconstructed image patch is a pixel-wise summation of two intermediate reconstructed image patches: the background and the foreground. The background is reconstructed from a set of small feature maps (background feature map) that can only encode large scale color and texture. The foreground is reconstructed from a set of crosswise sparse feature maps (foreground feature map). The foreground maps capture local high frequency signal: nuclei. We define *crosswise sparsity* as follows: when there is no detected nucleus at a location, neurons in all foreground feature maps at the same location should not be activated. The details of network parts 1–8 are in Tab. 1 and Tab. 2.

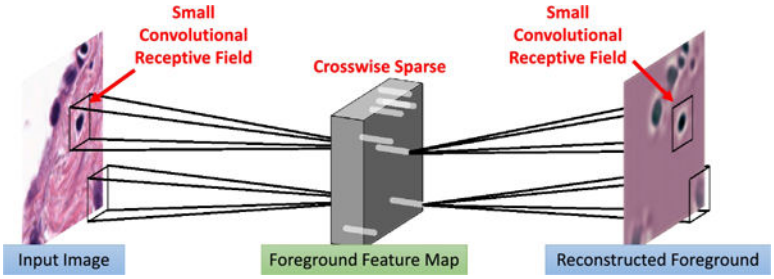


Figure 3:

An illustration of how each nucleus is encoded and reconstructed. First, the foreground feature map must be crosswise sparse (Eq. 2). Second, the size of the receptive field of each encoding neuron should be small enough that it contains only one nucleus in most cases.

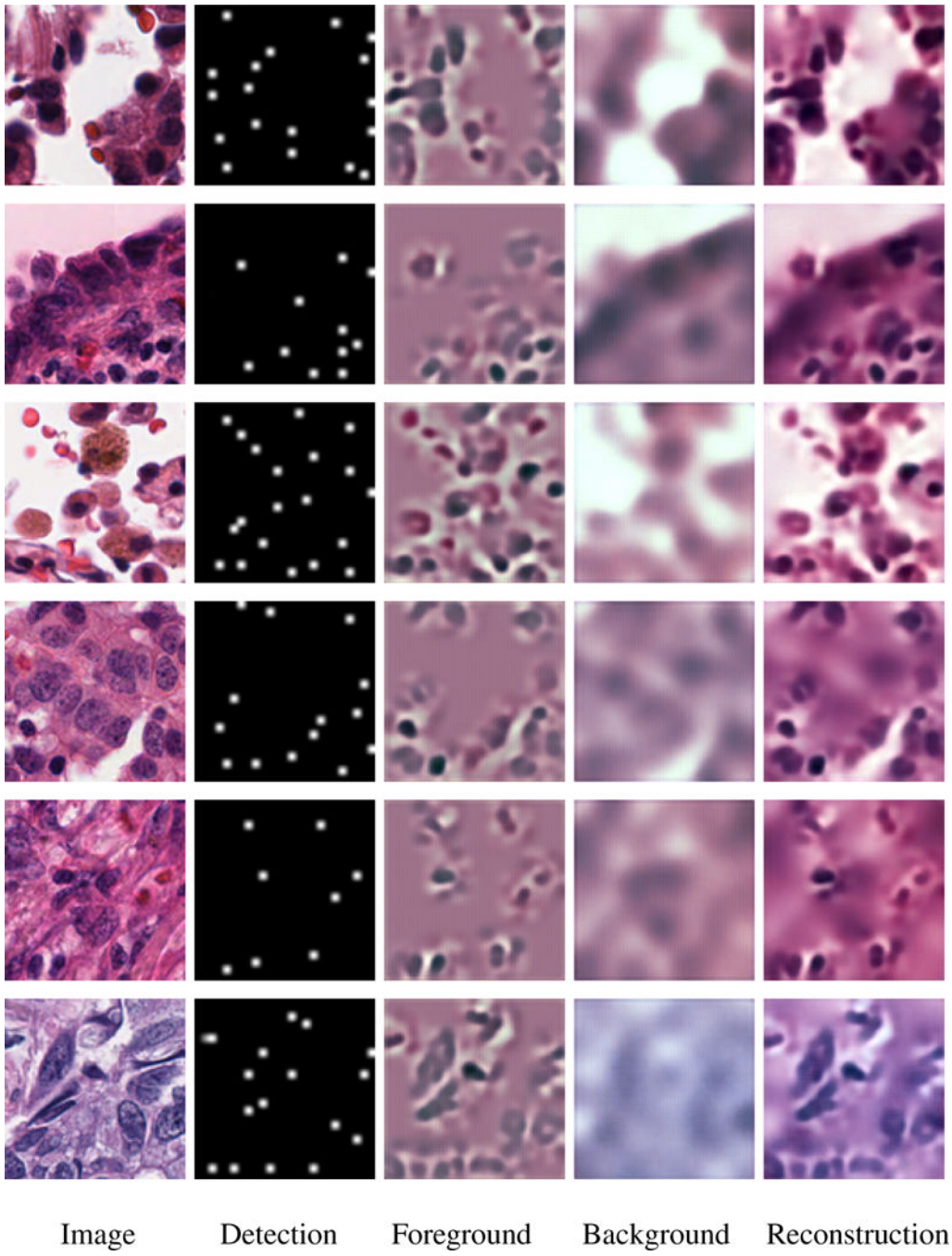


Figure 4:
Randomly selected examples of unsupervised nucleus detection representation results. Detection: the detection map. Foreground/Background: reconstructed foreground/background image. Reconstruction: the final reconstructed image.

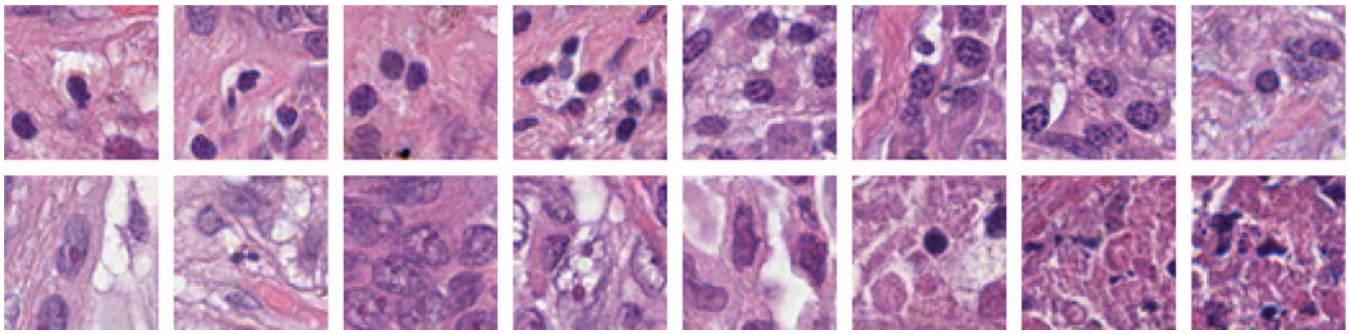


Figure 5:
Randomly selected examples of our self-collected lymphocyte classification dataset. Top row: image patches with a lymphocyte in the center (positive class). Bottom row: image patches with a nonlymphocyte object in the center (negative class).

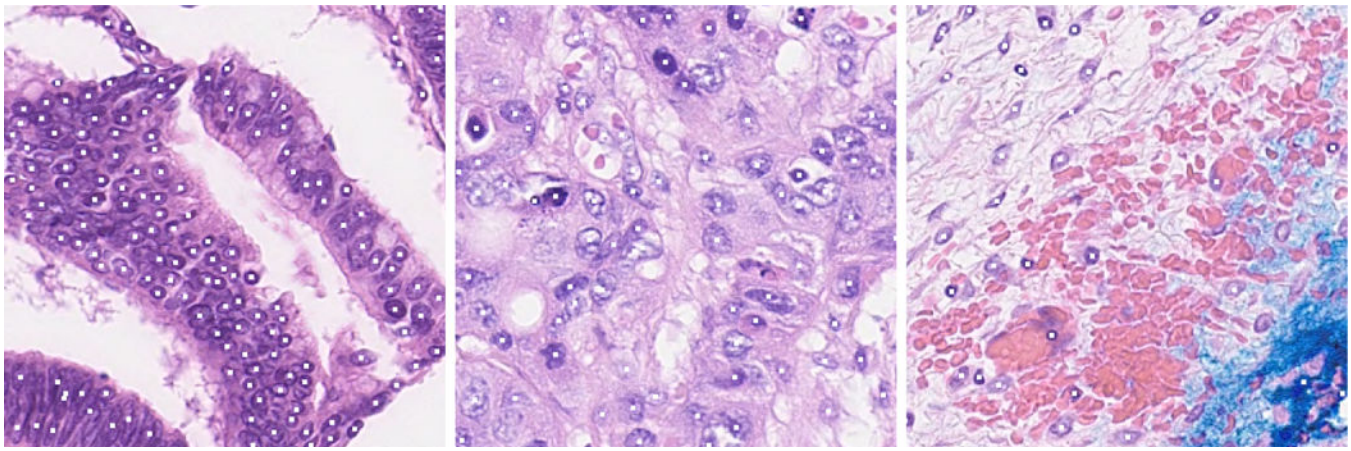


Figure 6:
Randomly selected examples of nucleus detection using our CSP-CNN, on the CRCHistoPheno- types nucleus detection dataset [34] (best viewed in color).

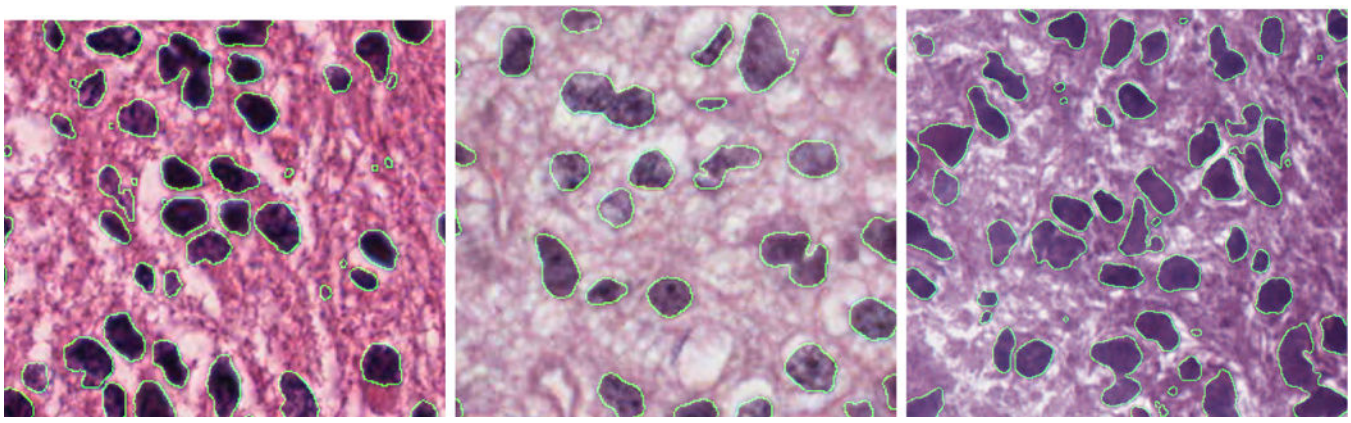


Figure 7:
Randomly selected examples of nucleus segmentation using our CSP-CNN, on the MICCAI 2015 nucleus segmentation challenge dataset (best viewed in color). The segmentation boundaries are in green.

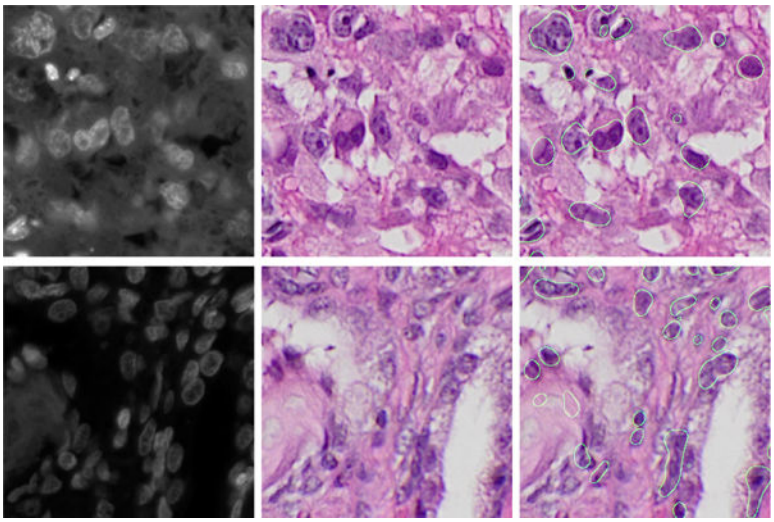


Figure 8:
Randomly selected examples of DAPI stained image patches (left) with corresponding H&E stained image patches (center) after image registration. The weak segmentation labels are displayed on the right using green contours.

Table 1:

The encoding layers in our CAE. Please refer to Fig. 2 for the overall network architecture. We apply batch normalization [57] before the leaky ReLU activation function [59] in all layers.

Part	Layer	Kernel size	Stride	Output size
1	Input	-	-	$100^2 \times 3$
	Convolution	5×5	1	$100^2 \times 100$
	Convolution	5×5	1	$100^2 \times 120$
	Average Pooling	2×2	2	$50^2 \times 120$
	Convolution	3×3	1	$50^2 \times 240$
	Convolution	3×3	1	$50^2 \times 320$
	Average Pooling	2×2	2	$25^2 \times 320$
	Convolution	3×3	1	$25^2 \times 640$
	Convolution	3×3	1	$25^2 \times 1024$
2	Convolution	1×1	1	$25^2 \times 100$
	Convolution	1×1	1	$25^2 \times 1$
3	Convolution	1×1	1	$25^2 \times 640$
	Convolution	1×1	1	$25^2 \times 100$
4	Convolution	1×1	1	$25^2 \times 128$
	Average Pooling	5×5	5	$5^2 \times 128$
	Convolution	3×3	1	$5^2 \times 64$
	Convolution	1×1	1	$5^2 \times 5$
5	Thresholding	Defined by Eq. 6		$25^2 \times 1$
6	Element-wise multiplication	Defined by Eq. 5		$25^2 \times 100$

Table 2:

The decoding layers in our CAE. Please refer to Fig. 2 for the overall network architecture. We apply batch normalization [57] before the leaky ReLU activation function [59] in all layers.

Part	Layer	Kernel size	Stride	Output size
7	Deconvolution	3×3	1	$25^2 \times 1024$
	Deconvolution	3×3	1	$25^2 \times 640$
	Deconvolution	4×4	0.5	$50^2 \times 640$
	Deconvolution	3×3	1	$50^2 \times 320$
	Deconvolution	3×3	1	$50^2 \times 320$
	Deconvolution	4×4	0.5	$100^2 \times 320$
	Deconvolution	5×5	1	$100^2 \times 120$
	Deconvolution	5×5	1	$100^2 \times 100$
	Deconvolution	1×1	1	$100^2 \times 3$
8	Deconvolution	3×3	1	$5^2 \times 256$
	Deconvolution	3×3	1	$5^2 \times 128$
	Deconvolution	9×9	0.2	$25^2 \times 128$
	Deconvolution	3×3	1	$25^2 \times 128$
	Deconvolution	3×3	1	$25^2 \times 128$
	Deconvolution	4×4	0.5	$50^2 \times 128$
	Deconvolution	3×3	1	$50^2 \times 64$
	Deconvolution	3×3	1	$50^2 \times 64$
	Deconvolution	4×4	0.5	$100^2 \times 64$
	Deconvolution	5×5	1	$100^2 \times 32$
	Deconvolution	5×5	1	$100^2 \times 32$
	Deconvolution	1×1	1	$100^2 \times 3$

Table 3:

Classification results measured by AUROC on two nucleus classification tasks described in Sec. 5.6.1. The proposed CSP-CNN outperforms the other methods significantly. Comparing the results of SP-CNN and our CSP-CNN, we see that the proposed crosswise constraint boosts performance significantly. Even with only 5% labeled training data, our CSP-CNN (5% data) outperforms other methods on the first dataset. The CSP-CNN (5% data) fails on the second dataset because when only using 5% training data, 5 out of 15 classes have less than 2 positive training instances which are too few for CNN training.

Methods	Nucleus Classification Datasets	
	Lymphocyte Classification	Nuclear Attribute & Shape [43]
SUP-CNN	0.4936	0.8487
SUP-CSP-CNN	0.5024	0.8480
DEN-CNN	0.5576	0.8656
SP-CNN	0.6262	0.8737
CSP-CNN	**0.7856**	**0.8788**
CSP-CNN (5% data)	0.7135	0.7128
Unsupervised features [70]	0.7132	-
Semi-supervised CNN [43]	-	0.8570
VGG16 [67]	0.6925	0.8480

Table 4:

Nucleus detection results on the CRCHistoPhenotypes nucleus detection dataset [34]. We achieved state-of-the-art results on this dataset. Even with no supervision, our crosswise sparse CAE (CSP-CAE) trained on lung adenocarcinoma image patches outperforms supervised methods trained on the CRCHistoPhenotypes colorectal adenocarcinoma dataset.

Methods	Precision	Recall	F-measure
SUP-CNN	0.7779	0.8921	0.8311
SUP-CSP-CNN	0.7625	0.8910	0.8218
DEN-CNN	0.7806	0.8625	0.8195
SP-CNN	0.8182	0.8268	0.8225
CSP-CNN	0.7883	0.8864	**0.8345**
CSP-CNN (5% data)	0.7349	0.8764	0.7994
CSP-CAE (fully unsupervised)	0.5796	0.6572	0.6159
U-net [14]	0.7681	0.8814	0.8209
Spatially Constraint CNN [34]	0.781	0.823	0.802
Structural Regression CNN [31]	0.783	0.804	0.793
Stacked Sparse Autoencoder + Softmax [40]	0.617	0.644	0.630
Local isotropic phase symmetry measure [40]	0.725	0.517	0.604
CRImage (morphological features) [71]	0.657	0.461	0.542

Table 5:

We evaluate the performance of the unsupervised nucleus detection with the CRCHistoPhenotypes nucleus detection dataset [34] with different sparsity rates p. The fully unsupervised detection results are comparable to many supervised methods. We can see that $p = 1.2$ and $p = 2.0$ yield similar results as $p = 1.6$ which is used in all other experiments.

Methods	Precision	Recall	F-measure
CSP-CAE (fully unsupervised) with $p = 1.2$	0.6141	0.6010	0.6075
CSP-CAE (fully unsupervised) with $p = 1.6$	0.5796	0.6572	0.6159
CSP-CAE (fully unsupervised) with $p = 2.0$	0.S298	0.6698	0.5916
CSP-CNN	0.7883	0.8864	**0.8345**
Spatially Constraint CNN [34]	0.781	0.823	0.802
Structural Regression CNN [31]	0.783	0.804	0.793
Stacked Sparse Autoencoder + Softmax [40]	0.617	0.644	0.630
Local isotropic phase symmetry measure [40]	0.725	0.517	0.604
CRImage (morphological features) [71]	0.657	0.461	0.542

Table 6:

To evaluate the unsupervised nucleus representation features, we use the features with a Multi-Layer Perceptron for lymphocyte classification. We name the method CSP-CAE-MLP. Our unsupervised features yield significantly better performance than baseline methods. Additionally, we test the CSP-CAE-MLP and the CSP-CNN with different sparsity rates p. Notice that $p = 1.2$ and $p = 2.0$ yield similar results as $p = 1.6$ which is used in all other experiments.

Methods	AUROC
CSP-CAE-MLP with $p = 1.2$	0.7536
CSP-CAE-MLP with $p = 1.6$	0.7591
CSP-CAE-MLP with $p = 2.0$	0.7410
CSP-CNN with $p = 1.2$	0.7714
CSP-CNN with $p = 1.6$	0.7856
CSP-CNN with $p = 2.0$	0.7841
Unsupervised features [70]	0.7132
VGG16 [67]	0.6925

Table 7:

Nucleus segmentation results on the MICCAI 2015 nucleus segmentation challenge dataset. Our CSP-CNN outperforms the highest challenge score which is a DICE-average of 0.80, even with only 5% of the sliding windows during training. We do not use pooling layers nor strided convolutional layers. Those layers discard important spatial information, because the size of nuclei are only around 20 × 20 pixels.

Methods	DICE-average
SUP-CNN	0.8216
SUP-CSP-CNN	0.8010
DEN-CNN	0.8235
SP-CNN	0.8338
CSP-CNN	**0.8362**
CSP-CNN (5% data)	0.8205
Contour-aware net (challenge winner) [33]	0.812
U-net [14]	0.7942